

What is claimed is:

1. A method for controlling an application process in a
5 distributed system, comprising:
 - providing at least one client;
 - providing at least one server, wherein
 - the system is organized according to a multi-tier model
 - and includes at least one first presentation layer, a second
 - 10 layer and a third data layer, and
 - the second layer is organized as a microkernel-based
 - client/server system; and
 - providing an interface formed as a message is specified
 - between the first and second layer, with a server request
 - 15 including at least the following:
 - the client translates the server request into a message
 - with the relevant arguments,
 - the client sends the message to the server,
 - the request is forwarded and processed in full, and
 - 20 the result of the request is returned on the basis of
 - the message.
2. The method according to claim 1, wherein the second layer
- and/or its components are configured for routing the server
- 25 request.
3. The method according to claim 1, wherein the server
- request is subdivided into a first transaction, originating
- from the client to the server, and a second transaction,
- 30 originating from the server to the client, and which are
- physically separate.
4. The method according to claim 1, wherein the client
- belongs to the first layer and the server belongs to the
- 35 second and/or third layer.

5. The method according to claim 1, wherein the server, after processing the request, sends a result of the request to the calling client or to a different destination address.

5 6. The method according to claim 1, wherein addresses and/or return addresses for the server request are coded in the message.

7. The method according to claim 1, wherein the microkernel
10 includes subsystems which belong to the second layer and/or third layer.

8. The method according to claim 1, wherein the server request nested, server requests.

15 9. The method according to claim 1, wherein a result of the request is returned on the basis of routing information included in the message.

20 10. The method according to claim 1, wherein the message comprises at least the following:
origin, in which the address of the client is coded,
source name, in which the address of the server to be called is coded, and
25 destination name, in which the return address for the result is coded.

11. The method according to claim 10, wherein origin and destination name correspond or are different.

30 12. The Method according to claim 1, wherein the message of the client from the first layer is sent to a root component of the microkernel of the second layer, which then forwards the message to a processing component.

35

13. The method according to claim 1, wherein the processing in the second layer is performed asynchronously with respect to the processing in the first and/or third layer.
- 5 14. The method according to claim 1, wherein one part of the client which communicates with the second layer is blocked for a time between the server call and transmission of the message or receipt of a confirmation.
- 10 15. The method according to claim 1, wherein multiple calls of multiple clients are stored in a queue which operates according to a FIFO principle.
- 15 16. The method according to claim 1, wherein the microkernel of the second layer includes multiple subsystems which are subdivided into one or more components.
17. The method according to claim 1, wherein the servers of the second layer is/are not required to administer
20 request-related address information.
18. A client/server system for controlling and/or implementing an application process, comprising:
at least one client;
25 at least one server, wherein
the system is organized according to a multi-tier model and includes at least one first presentation layer, a second layer and a third data layer, and
the second layer is organized as a microkernel-based
30 client/server system; and
an interface formed as a message is specified between the first and second layer, with a server request including at least the following:
the client translates the server request into a message
35 with the relevant arguments,
the client sends the message to the server,
the request is forwarded and processed in full, and

the result of the request is returned on the basis of the message.

5 20. A computer program product having a computer-readable medium with computer program code elements, in which, after the computer program has been loaded, the computer causes the program to execute the following:

organizing the system according to a multi-tier model
10 and including at least one first presentation layer, a second layer and a third data layer; and

organizing the second layer as a microkernel-based client/server system; and

providing a message specified between the first and
15 second layer, with a server request including at least the following:

translating the server request into a message with the relevant arguments,

sending the message to the server,

20 forwarding and processing the request, and

returning a result of the request on the basis of the message.

21. A device for executing and/or organizing an application
25 process in a distributed system, comprising:

at least one client;

at least one server, in which the system is organized according to a multi-tier model and comprises at least a first presentation layer, a second layer and a third data
30 layer, wherein

the second layer is organized as a microkernel-based client/server system; and

an interface, in the form of a message, is arranged between the first and second layer, where

35 the device for processing a server request comprises at least the following:

a message generation module which is configured to convert the server request into a message with relevant arguments,

5 a send module which is configured to send the message to the server,

a processing unit which forwards and processes the request, and

a return module which sends the result of the request back on the basis of the message.

10

22. The device according to claim 21, wherein the device has a computer architecture which is configured to execute the method in which the device executes:

15 organizing the system according to a multi-tier model and including at least one first presentation layer, a second layer and a third data layer; and

organizing the second layer as a microkernel-based client/server system; and

20 providing a message specified between the first and second layer, with a server request including at least the following:

translating the server request into a message with the relevant arguments,

25 sending the message to the server,

forwarding and processing the request, and

returning a result of the request on the basis of the message.